

# STRING

## 1. What is a string ?

A string is a sequence of characters that are enclosed within single quotes or double quotes.

Example: "Python is a very simple language." or 'Python is a very simple language'

## 2. How to define a string ?

Syntax to define a string is as follows:

```
stringName = "string"
```

or

```
stringName = 'string'
```

Example: `programmingLanguage = "Python"`

## 3. How are strings stored ?

Strings are stored as an indexed sequence of characters. The starting index is zero.

|               |   |   |   |   |   |   |
|---------------|---|---|---|---|---|---|
| <b>String</b> | P | Y | T | H | O | N |
| <b>Index</b>  | 0 | 1 | 2 | 3 | 4 | 5 |

## 4. What are some of the string operations ?

Following are some of the string operations that can be performed in Python. Consider the following example string which we will be using for further explanation.

```
welcomeMessage = "Welcome to this Introduction  
course on Python programming. Hope you are having  
a good time learning."
```

### 1. Upper

Converts all the characters of a string to uppercase.

Syntax: `stringName.upper()`

Example: `"welcomeMessage.upper()"`

returns 'WELCOME TO THIS INTRODUCTION COURSE ON

```
PYTHON PROGRAMMING. HOPE YOU ARE HAVING A GOOD  
TIME LEARNING.'
```

## 2. Lower

Converts all the characters of a string to lowercase.

Syntax: `stringName.lower()`

**Example:** `welcomeMessage.lower()` returns  
'welcome to this introduction course on python  
programming. hope you are having a good time  
learning.'

## 3. Replace

Replaces the specified sequence of character(s) in the first parameter with that of the second parameter.

Syntax: `stringName.replace("sequenceToBeReplaced", "sequenceToReplace")`

**Example:**

`welcomeMessage.replace("learning", "experimenting")` returns 'welcome to this  
introduction course on python programming. hope  
you are having a good time experimenting.'

## 4. Slicing

Returns a slice of the string based on the indices passed.

Syntax: `stringName[startingIndex:endingIndex]`

**Note:**

1. Characters starting from index "startingIndex" upto "endingIndex - 1" will be printed

**Example:** `welcomeMessage[0:7]` returns  
"welcome"

2. If startingIndex is not mentioned, characters starting from 0 to "endingIndex - 1" will be printed

**Example:** `welcomeMessage[:57]` returns 'welcome  
to this introduction course on python  
programming'

3. If `endingIndex` is not mentioned, characters starting from `startingIndex` to end of the string will be printed

**Example:** `welcomeMessage[59:]` returns `'hope you are having a good time experimenting.'`

4. If both `startingIndex` and `endingIndex` are not mentioned, the whole string will be printed

**Example:** `welcomeMessage[:]` returns `'welcome to this introduction course on python programming. hope you are having a good time experimenting.'`

## 5. Length

Returns length of a string

Syntax: `len(string)`

**Example:** `len(welcomeMessage)` returns `100`

## 5. Basic string formatting in python

Consider the following string.

```
programmingLanguage = "Python"
```

If you need to include this string as a part of another string, you need to use a set of opening and closing curly braces as a placeholder `'{'`

**Example:** `print "Welcome to {} programming".format(programmingLanguage)`

This would print `"Welcome to Python programming."`